

II. AMENDMENTS

IN THE CLAIMS

- 1 1. (currently amended) A method of ~~maintaining~~ facilitating maintenance of an
2 index during a reorganization of data in a database, wherein said database comprises one or
3 more records and each said record includes a root segment, the method comprising:
4 storing each root segment in a separate storage location; and
5 retaining each root segment in its storage location during a reorganization of data in
6 said database.
- 1 2. (original) The method of claim 1, wherein all root segments stored within a
2 block of storage locations are stored in contiguous storage locations.
- 1 3. (original) The method of claim 1, wherein each said storage location which
2 stores a root segment is fixed.
- 1 4. (original) The method of claim 2, wherein each said storage location which
2 stores a root segment is fixed.
- 1 5. (original) The method of claim 1, wherein said database is an IMS full
2 function database.
- 1 6. (original) The method of claim 3, wherein each said root segment is stored in
2 a fixed storage location at the time it is added to the database.
- 1 7. (currently amended) A method of ~~maintaining~~ facilitating maintenance of an
2 index during a reorganization of data in a database, wherein said database comprises one or
3 more records and each said record includes a root segment and one or more non-root
4 segments, the method comprising:
5 storing each root segment in a separate storage location, wherein each storage location
6 is associated with a particular block of storage locations;
7 retaining each root segment in its storage location during a reorganization of data in
8 said database; and

9 storing each non-root segment, associated with a first root segment, in a block of
10 storage locations in which said first root segment is also stored.

1 8. (original) The method of claim 7, wherein all root segments stored within a
2 block of storage locations are stored in contiguous storage locations.

1 9. (original) The method of claim 7, wherein each said storage location which
2 stores a root segment is fixed.

1 10. (original) The method of claim 8, wherein each said storage location which
2 stores a root segment is fixed.

1 11. (original) The method of claim 7, wherein said database is an IMS full
2 function database.

1 12. (original) The method of claim 9, wherein each said root segment is stored in
2 a fixed storage location at the time it is added to the database.

1 13. (currently amended) A method of ~~maintaining~~ facilitating maintenance of an
2 index during a reorganization of data in a database, wherein said database comprises one or
3 more records, each said record includes a root segment and one or more non-root segments,
4 and each root segment comprises a prefix component and a data component, the method
5 comprising:

6 storing said prefix component of each root segment in a separate storage location; and
7 retaining said prefix component of each root segment in its storage location during a
8 reorganization of data in said database.

1 14. (original) The method of claim 13, wherein all root segment prefix
2 components stored within a block of storage locations are stored in contiguous storage
3 locations.

1 15. (original) The method of claim 13, wherein each said storage location which
2 stores a root segment prefix component is fixed.

1 16. (original) The method of claim 14, wherein each said storage location which
2 stores a root segment prefix component is fixed.

1 17. (original) The method of claim 13, wherein said database is an IMS full
2 function database.

1 18. (original) The method of claim 15, wherein each said root segment prefix
2 component is stored in a fixed storage location at the time it is added to the database.

1 19. (currently amended) A method of ~~maintaining~~ facilitating maintenance of an
2 index during a reorganization of data in a database, wherein said database comprises one or
3 more records, each said record includes a root segment and one or more non-root segments,
4 and each root and non-root segment comprises a prefix component and a data component, the
5 method comprising:
6 storing said prefix component of each root segment in a separate storage location,
7 wherein each storage location is associated with a particular block of storage
8 locations;
9 retaining said prefix component of each root segment in its storage location during a
10 reorganization of data in said database; and
11 storing said prefix component of each non-root segment which is associated with a
12 first root segment, in a block of storage locations in which said prefix
13 component of said first root segment is also stored.

1 20. (original) The method of claim 19, wherein all root segments stored within a
2 block of storage locations are stored in contiguous storage locations.

1 21. (original) The method of claim 19, wherein said database is an IMS full
2 function database.

1 22. (original) A method of facilitating correction of an index after a
2 reorganization of data in a database, wherein said index comprises index entries, said
3 database comprises one or more records, each said record comprises one or more segments,
4 and each index entry comprises an address to a target segment, the method comprising:

5 prior to a reorganization of data in said database, assigning a unique token to each
6 target segment and each corresponding index entry having an address to a
7 target segment, wherein said unique token for a given target segment and for a
8 corresponding index entry is the same;
9 after a reorganization of data in said database, reading the unique token of a first
10 index entry;
11 reading the unique token of each target segment until a match is found between the
12 unique token of a matching target segment and the unique token of said first
13 index entry;
14 determining the address of said matching target segment; and
15 replacing the address of said first index entry with the address of said matching target
16 segment.

1 23. (original) The method of claim 19, wherein said index which is to be
2 corrected is a secondary index and each said segment is a non-root segment.

1 24. (original) The method of claim 22, wherein said database is an IMS database.

1 25. (original) The method of claim 23, wherein said database is an IMS database.

1 26. (original) A method of facilitating correction of an index after a
2 reorganization of data in a database, wherein said index comprises index entries, said
3 database comprises one or more records, each said record comprises a root segment and one
4 or more non-root segments, and each index entry comprises an address to a target segment
5 included within said root and non-root segments, the method comprising:
6 prior to a reorganization of data in said database, assigning a unique token to each
7 target segment and each corresponding index entry having an address to a
8 target segment, wherein said unique token for a given target segment and for a
9 corresponding index entry is the same;
10 after a reorganization of data in said database, reading the unique token of a first
11 index entry for a first record;
12 reading the unique token of each non-root segment within said first record until a
13 match is found between the unique token of a matching target segment and the
14 unique token of said first index entry;

15 determining the address of said matching target segment; and
16 replacing the address of said first index entry with the address of said matching target
17 segment.

1 27. (original) The method of claim 26, wherein said index which is to be
2 corrected is a secondary index.

1 28. (original) The method of claim 26, wherein said database is an IMS database.

1 29. (original) The method of claim 27, wherein said database is an IMS database.

1 30. (original) The method of claim 26, wherein each said unique token includes
2 one or more the following: (i) a born on date of the target segment to which said unique
3 token is assigned; or (ii) a key field of the target segment to which said unique token is
4 assigned.

1 31. (original) A method of facilitating correction of an index after a
2 reorganization of data in a database, wherein said index comprises index entries, said
3 database comprises one or more records, each said record comprises a root segment and one
4 or more non-root segments, and each index entry comprises an address to a target segment
5 included within said root and non-root segments, wherein said root segment and one or more
6 non-root segments for a record are stored within a block of storage locations, the method
7 comprising:

8 prior to a reorganization of data in said database, assigning a unique token to each
9 target segment and each corresponding index entry having an address to a
10 target segment, wherein said unique token for a given target segment and for a
11 corresponding index entry is the same;
12 after a reorganization of data in said database, reading the unique token of a first
13 index entry for a first record;
14 reading the unique token of each non-root segment within said first record until a
15 match is found between the unique token of a matching target segment and the
16 unique token of said first index entry;
17 determining the address of said matching target segment; and

18 replacing the address of said first index entry with the address of said matching target
19 segment.

1 32. (original) The method of claim 31, wherein said index which is to be
2 corrected is a secondary index.

1 33. (original) The method of claim 31, wherein said database is an IMS database.

1 34. (original) The method of claim 32, wherein said database is an IMS database.

1 35. (original) The method of claim 31, wherein each said root segment is stored
2 in a fixed storage location prior to a reorganization of data in said database, and said root
3 segment is retained in said fixed storage location during a reorganization.

1 36. (original) The method of claim 31, wherein each said unique token is includes
2 one or more the following: (i) a born on date of the target segment to which said unique
3 token is assigned; or (ii) a key field of the target segment to which said unique token is
4 assigned.

1 37. (original) A method of facilitating correction of an index after a
2 reorganization of data in a database, wherein said index comprises index entries, said
3 database comprises one or more records, each said record comprises a root segment and one
4 or more non-root segments, each root and non-root segment comprises a prefix component
5 and a data component, and each index entry comprises an address to the prefix component of
6 a target segment included within said root and non-root segments, the method comprising:
7 prior to a reorganization of data in said database, assigning a unique token to the
8 prefix component of each target segment and each corresponding index entry
9 having an address to the prefix component of a target segment, wherein said
10 unique token for the prefix component of a given target segment and for a
11 corresponding index entry is the same;
12 after a reorganization of data in said database, reading the unique token of a first
13 index entry for a first record;
14 reading the unique token of the prefix component of each non-root segment within
15 said first record until a match is found between the unique token of a matching

16 target segment prefix component and the unique token of said first index
17 entry;
18 determining the address of said matching target segment prefix component; and
19 replacing the address of said first index entry with the address of said matching target
20 segment prefix component.

1 38. (original) The method of claim 37, wherein said index which is to be
2 corrected is a secondary index.

1 39. (original) The method of claim 37, wherein said database is an IMS database.

1 40. (original) The method of claim 37, wherein each said root segment is stored
2 in a fixed storage location prior to a reorganization of data in said database, and said root
3 segment is retained in said fixed storage location during a reorganization.

1 41. (original) The method of claim 37, wherein each index entry and each non-
2 root target segment further comprise a root segment identifier which identifies what root
3 segment said non-root target segment is associated with, and each identified root segment
4 comprises addresses to all non-root segments, within a record, associated with said identified
5 root segment.

1 42. (original) The method of claim 41, wherein each said unique token includes
2 one or more the following: (i) a born on date of the target segment to which said unique
3 token is assigned; or (ii) a key field of the target segment to which said unique token is
4 assigned.

1 43. (original) The method of claim 42, wherein each said unique token for an
2 index entry and each non-root target segment further comprises said root segment identifier
3 which identifies what root segment said non-root target segment is associated with.

1 44. (original) The method of claim 41, wherein said database is an IMS database.

1 45. (original) The method of claim 41, wherein said index which is to be
2 corrected is a secondary index.

1 46. (original) The method of claim 37, wherein each said unique token includes
2 one or more the following: (i) a born on date of the target segment to which said unique
3 token is assigned; or (ii) a key field of the target segment to which said unique token is
4 assigned.

1 47. (original) The method of claim 22, wherein after a reorganization of data in
2 said database but before taking steps to correct said address of said first index entry,
3 determining if said address of said first index entry is valid and then correcting said address
4 only if it is invalid.

1 48. (original) The method of claim 47, wherein determining if said address of said
2 first index entry is valid comprises comparing the unique token of said first index entry to the
3 unique token associated with a segment located at said address, and ascertaining if said
4 unique tokens are the same.

1 49. (original) The method of claim 47, wherein determining if said address of said
2 first index entry is valid comprises comparing a segment code of said first index entry to a
3 segment code associated with a segment located at said address, and if said segment codes are
4 the same, then comparing the unique token of said first index entry to the unique token
5 associated with said segment located at said address, and ascertaining if said unique tokens
6 are the same.

1 50. (original) The method of claim 26, wherein after a reorganization of data in
2 said database but before taking steps to correct said address of said first index entry,
3 determining if said address of said first index entry is valid and then correcting said address
4 only if it is invalid.

1 51. (original) The method of claim 50, wherein determining if said address of said
2 first index entry is valid comprises comparing the unique token of said first index entry to the
3 unique token associated with a segment located at said address, and ascertaining if said
4 unique tokens are the same.

1 52. (original) The method of claim 50, wherein determining if said address of said
2 first index entry is valid comprises comparing a segment code of said first index entry to a
3 segment code associated with a segment located at said address, and if said segment codes are
4 the same, then comparing the unique token of said first index entry to the unique token
5 associated with said segment located at said address, and ascertaining if said unique tokens
6 are the same.

1 53. (original) The method of claim 31, wherein after a reorganization of data in
2 said database but before taking steps to correct said address of said first index entry,
3 determining if said address of said first index entry is valid and then correcting said address
4 only if it is invalid.

1 54. (original) The method of claim 53, wherein determining if said address of said
2 first index entry is valid comprises comparing the unique token of said first index entry to the
3 unique token associated with a segment located at said address, and ascertaining if said
4 unique tokens are the same.

1 55. (original) The method of claim 53, wherein determining if said address of said
2 first index entry is valid comprises comparing a segment code of said first index entry to a
3 segment code associated with a segment located at said address, and if said segment codes are
4 the same, then comparing the unique token of said first index entry to the unique token
5 associated with said segment located at said address, and ascertaining if said unique tokens
6 are the same.

1 56. (original) The method of claim 37, wherein after a reorganization of data in
2 said database but before taking steps to correct said address of said first index entry,
3 determining if said address of said first index entry is valid and then correcting said address
4 only if it is invalid.

1 57. (original) The method of claim 56, wherein determining if said address of said
2 first index entry is valid comprises comparing the unique token of said first index entry to the
3 unique token associated with a prefix component of a segment located at said address, and
4 ascertaining if said unique tokens are the same.

1 58. (original) The method of claim 56, wherein determining if said address of said
2 first index entry is valid comprises comparing a segment code of said first index entry to a
3 segment code associated with a prefix component of a segment located at said address, and if
4 said segment codes are the same, then comparing the unique token of said first index entry to
5 the unique token associated with said prefix component of a segment located at said address,
6 and ascertaining if said unique tokens are the same.

1 59. (original) A program storage media readable by a machine and containing
2 instructions for performing the method contained in claim 1.

1 60. (original) A program storage media readable by a machine and containing
2 instructions for performing the method contained in claim 7.

1 61. (original) A program storage media readable by a machine and containing
2 instructions for performing the method contained in claim 13.

1 62. (original) A program storage media readable by a machine and containing
2 instructions for performing the method contained in claim 19.

1 63. (original) A program storage media readable by a machine and containing
2 instructions for performing the method contained in claim 22.

1 64. (original) A program storage media readable by a machine and containing
2 instructions for performing the method contained in claim 26.

1 65. (original) A program storage media readable by a machine and containing
2 instructions for performing the method contained in claim 31.

1 66. (original) A program storage media readable by a machine and containing
2 instructions for performing the method contained in claim 37.